

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Lenč Skumavec

**Izdelava spletne aplikacije za  
ocenjevanje po metodi 360°**

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM  
PRVE STOPNJE  
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Luka Šajn

Ljubljana, 2017

COPYRIGHT. Rezultati diplomske naloge so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavo in koriščenje rezultatov diplomske naloge je potrebno pisno privoljenje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

*Besedilo je oblikovano z urejevalnikom besedil  $\text{\LaTeX}$ .*

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

V današnjem času nam s pomočjo sodobnih tehnologij in svetovnega spleta uspeva poenostaviti veliko časovno potratnih procesov in kompleksnih izračunov. Eden izmed časovno potratnih procesov, ki vključujejo kompleksne izračune je tudi ocenjevanje zaposlenih po metodi 360°, ki je znano tudi kot metoda več ocenjevalcev ali povratna informacija iz več virov. Izdelajte spletno aplikacijo za ocenjevanje zaposlenih znotraj posameznega podjetja oziroma organizacije po metodi 360°. Aplikacija naj bo prilagojena tako za spletne brskalnike na namiznih in prenosnih računalnikih kot tudi za mobilne naprave. Za izdelavo aplikacije si izberite ustrezno tehnologijo. Zagotovite tudi varnost podatkov in varno uporabo aplikacije, saj so spletne aplikacije priljubljena tarča hekerskih vdorov.



*Rad bi se zahvalil svojim staršem in ostalim sorodnikom ter prijateljem, ki so me podpirali tekom študija in med izdelavo diplomske naloge. Zahvala gre tudi podjetjema ProActa, d. o. o., in ResEvo, d. o. o., ki sta mi omogočila uporabo aplikacije za izdelavo diplomske naloge ter mi pomagala pri sami izvedbi. Posebej bi se rad zahvalil tudi mentorju doc. dr. Luki Šajnu za vso pomoč in uporabne nasvete pri izdelavi diplomske naloge.*



# Kazalo

Povzetek

Abstract

<b>1</b>	<b>Uvod</b>	<b>1</b>
<b>2</b>	<b>Opis analize 360°</b>	<b>5</b>
2.1	Koristi in pasti uvajanja analize 360° . . . . .	6
2.2	Kako deluje . . . . .	7
2.3	Ocenjevanje kompetenc . . . . .	8
<b>3</b>	<b>Pregled programskih tehnologij in orodij za razvoj aplikacije</b>	<b>11</b>
3.1	Razvojna orodja . . . . .	11
3.2	Opis uporabljenih tehnologij . . . . .	14
<b>4</b>	<b>Opis funkcionalnosti in potek aplikacije</b>	<b>17</b>
4.1	Skrbnik sistema . . . . .	17
4.1.1	Prijava v aplikacijo in ponastavitev gesla . . . . .	17
4.1.2	Nadzorna plošča . . . . .	18
4.1.3	Vnos in urejanje šifrantov . . . . .	20
4.1.4	Preverjanje pravilnosti vnesenih elektronskih naslovov .	21
4.1.5	Določanje relacij med uporabniki in vnos spremnega besedila . . . . .	21
4.1.6	Urejanje vprašalnikov . . . . .	23
4.1.7	Izpis poročil . . . . .	23

4.2	Uporabnik . . . . .	24
4.2.1	Prijava v aplikacijo in ponastavitev gesla . . . . .	24
4.2.2	Ocenjevanje vprašalnikov . . . . .	25
<b>5</b>	<b>Potek in razvoj ključnih rešitev</b>	<b>27</b>
5.1	Namestitev in struktura ogrodja Laravel . . . . .	27
5.2	Načrtovanje in podatkovne baze . . . . .	29
5.3	Registracija in prijava uporabnikov . . . . .	33
5.4	Avtorizacija uporabnikov . . . . .	35
5.5	Vnos in urejanje šifrantov . . . . .	36
5.6	Preverjanje pravilnosti elektronskih naslovov . . . . .	37
5.7	Relacije med zaposlenimi in vnos spremnega besedila . . . . .	38
5.8	Urejanje vprašalnikov . . . . .	39
5.9	Ocenjevanje . . . . .	40
5.10	Izdelava poročil . . . . .	41
<b>6</b>	<b>Sklepne ugotovitve</b>	<b>43</b>
	<b>Literatura</b>	<b>46</b>





# Seznam uporabljenih kratic

kratica	angleško	slovensko
<b>PHP</b>	Personal Home Page	Odprtokodni programski jezik namenjen za uporabo na spletnih strežnikih
<b>HTTP</b>	Hypertext Transfer Protocol	Protokol za prenos informacij na spletu
<b>MVC</b>	Model-View-Controller	Način zasnove arhitekture aplikacije
<b>HTML</b>	Hyper Text Markup Language	Označevalni jezik za izdelovanje spletnih strani
<b>CSS</b>	Cascading Style Sheets	Jezik za oblikovanje spletnih strani
<b>SQL</b>	Structured Query Language	Programski jezik za delo s podatkovnimi bazami
<b>AJAX</b>	Asynchronous JavaScript and XML	Asinhroni JavaScript in XML
<b>PDF</b>	Portable Document Format	Standard za izmenjavo elektronskih dokumentov
<b>JSON</b>	JavaScript Object Notation	Format za izmenjavo podatkov
<b>API</b>	Application programming interface	Kratica za aplikacijski programski vmesnik
<b>MX</b>	Mail exchanger record	Kratica za protokol, ki preslika domensko ime v spisek strežnikov za določeno domeno

# Povzetek

**Naslov:** Izdelava spletne aplikacije za ocenjevanje po metodi 360°

**Avtor:** Lenč Skumavec

V sklopu diplomske naloge smo za potrebe podjetja Pro Acta, d. o. o., izdelali spletno aplikacijo za ocenjevanje zaposlenih v posameznem podjetju ali organizaciji po metodi ocenjevanja 360 stopinj. Ocenjevanje po metodi 360 stopinj, ki je znana tudi kot metoda več ocenjevalcev ali povratna informacija iz več virov, je eden izmed glavnih trendov zadnjega desetletja na področju ravnanja z ljudmi. Aplikacijo smo izdelali z uporabo ogrodja Laravel ter nekaterih drugih odprtokodnih modulov in knjižnic. V prvem delu je opisano ocenjevanje po metodi 360 stopinj, kako deluje, kako jo izvajamo ter njene prednosti in slabosti. V drugem delu pa je opisan postopek načrtovanja in izdelave aplikacije, tehnologija in orodja, ki so bila uporabljena za izdelavo aplikacije, izdelava ključnih rešitev ter končni rezultat uporabe aplikacije.

**Ključne besede:** spletna aplikacija, ocenjevanje, analiza 360, Laravel, povratna informacija.



# Abstract

**Title:** Creating a Web Application For 360 ° Feedback

**Author:** Lenč Skumavec

For the needs of the company Pro Acta, d. o. o., we created as part of the thesis, a web application for evaluation of employees in an individual company or organization, using the assessment method 360 degree feedback. Evaluating by the 360 degree feedback, known also as multi-rater feedback or multi source feedback, is one of the main trends in the management of human resources. We created the application by using Laravel framework and some other open source modules and libraries. In the first part of the thesis, we described evaluation by using the 360 degree feedback method, how it works, how we perform it and also advantages and disadvantages of the method. In the second part, the procedure of planning and making the application, technology and tools that were used to build application, making key solutions and the final result of the use of application are described.

**Keywords:** web application, assessment, 360 degree feedback, Laravel.



# Poglavje 1

## Uvod

Danes se številni menedžerji in vodje podjetij ter organizacij zavedajo, da so njihovi zaposleni največje bogastvo, ki ga premorejo, le redki izmed njih pa to tudi dejansko upoštevajo. Na nek način se zavedajo, da zaposleni lahko naredijo razliko v podjetju in ustvarjajo dodano vrednost. Določeni pogoji morajo biti izpolnjeni, če želimo, da bodo zaposleni prispevali k rasti podjetja in ustvarjanju dodane vrednosti. Pravi ljudje morajo biti na pravih delovnih mestih. Na avtobusu lahko imamo prave ljudi, a je pomembno tudi to, da so na pravih sedežih. S pomočjo različnih analiz lahko ugotovimo, ali so pravi ljudje na pravih delovnih mestih. Ena najbolj razširjenih in pogosto uporabljenih metod v zadnjem času je analiza na osnovi metod 360°. [5] V sklopu diplomske naloge smo za podjetje Pro Acta, d. o. o., izdelali spletno aplikacijo za merjenje in ocenjevanje kompetenc znotraj posameznega podjetja ali organizacije na osnovi metode 360°.

Kljub temu da je danes na svetovnem spletu kar nekaj aplikacij, ki omogočajo izvedbo merjenja kompetenc po metodi ocenjevanja 360 stopinj, smo se odločili, da razvijemo lastno rešitev v obliki spletne aplikacije. Glavni vzrok za razvoj svoje aplikacije je, da večina že obstoječih rešitev ne ponuja dovolj podrobne analize kompetenc, ki je rezultat ocenjevanja sodelujočih. V današnjem času je uporaba spletnih aplikacij zelo razširjena in popularnejša od klasičnih namiznih programov. Za uporabo le-teh potrebujemo spletni br-

skalnik in povezavo s svetovnim spletom ali pa samo povezavo do lokalnega omrežja preko katerega je spletna aplikacija dostopna.

Za namestitev in delovanje spletne aplikacije, ki smo jo razvili, potrebujemo spletni strežnik, na katerem teče skriptni jezik PHP, in podatkovno bazo MySQL ali PostgreSQL. Razvoj lastne aplikacije je stroškovno in časovno velik zalogaj, zato smo se odločili, da bomo aplikacijo razvili s pomočjo ogrodja Laravel, kar nam je prihranilo veliko razvojnega časa, saj so komponente, kot sta avtentikacija in avtorizacija uporabnikov, že narejene in pripravljene za uporabo. Aplikacija, ki smo jo razvili ima dvouporabniško strukturo. Uporabniku je dodana vloga administratorja ali navadnega uporabnika. Administrator ima dostop do vseh možnosti aplikacije, preko katerih vnaša in ureja vse potrebne parametre za izvajanje ocenjevanja. Zaledni sistem administratorja je razdeljen v sedem večjih razdelkov. Ob prijavi v aplikacijo najprej prispe do nadzorne plošče, kjer ima na voljo celoten pregled nad vnesenimi parametri ter sodelujočimi uporabniki pri ocenjevanju.

Uporabniki imajo v vlogi uporabnika na voljo dostop samo do razdelka, kjer so jim prikazani napotki za ocenjevanje uporabnikov ter seznam do povezav vseh vprašalnikov za ocenjevanje svojih nadrejenih, sodelavcev, podrejenih ali samoocenjevanja. Ob zaključenem ocenjevanju vseh sodelujočih aplikacija za vsakega posameznika izdela izčrpno poročilo, ki je na voljo v PDF zapisu. Obseg posameznega poročila je od trideset do petdeset strani formata A4 in je razdeljeno na pet večjih poglavij. Aplikacija poročila izdela z uporabo označevalnega jezika HTML5, JavaScript knjižnico Highcharts in z uporabo naprednih medijskih poizvedb jezika CSS.

Po uvodnem poglavju diplomske naloge sledi drugo poglavje, v katerem je opisano ocenjevanje po metodi 360 stopinj. Opisane so prednosti in slabosti ocenjevanja, razlogi za izbiro ocenjevanja kompetenc po tej metodi ter opis, kako se ocenjevanje izvaja. V tretjem poglavju sledi opis orodij in tehnologij, s katerimi smo razvili spletno aplikacijo. V četrtem poglavju sledi podrobnejši opis in potek aplikacije. To poglavje vsebuje opis posameznih gradnikov aplikacije ter potek aplikacije z vidika administratorja in navadnih



uporabnikov. V petem poglavju je opisan razvoj ključnih rešitev spletne aplikacije ter je podan podrobnejši opis razvoja ključne funkcionalnosti aplikacije s pomočjo označevalnega jezika HTML5, medijskih poizvedb jezika CSS, ogrodja Laravel in uporabo jezika JavaScript in nekaterih JavaScript knjižnic. V zaključnem poglavju so opisane sklepne ugotovitve.



## Poglavje 2

### Opis analize 360°

Ena izmed najbolj razširjenih in najbolj pogosto uporabljenih analiz na področju ravnanja z ljudmi za merjenje in razvoj kompetenc v zadnjem desetletju je analiza na osnovi metode 360 stopinj [5], znana je tudi kot povratna informacija iz več virov ali metoda več ocenjevalcev [9]. V procesu se zaposlene v podjetjih in organizacijah ocenjuje po kompetencah, pomembnih za podjetje [5]. Kompetence so sposobnosti in zmožnosti, ki so potrebne, da nekdo uspešno in učinkovito opravi določeno delo ali vlogo. Poleg ocen posameznih kompetenc ocenjevalci podajo tudi svoja mnenja. Ocenjujejo jih njihovi nadrejeni, sodelavci, podrejeni, hkrati pa se tudi samoocenjujejo [5]. Če se podjetje na primer ukvarja s prodajo ali nabavo, je v ocenjevanje zaposlenih smiselno vključiti tudi zunanje virov, kot so stranke in dobavitelji, s katerimi udeleženci ocenjevanja tesno sodelujejo [9].

Ocenjevanje poteka preko posebnih vprašalnikov. Vsak ocenjevalec, ki je vključen v ocenjevanje, na podlagi lastnih vedenj in izkušenj izpolni izbor vprašanj, ki ocenjujejo potenciale določenega posameznika. Podatki, zbrani z ocenjevanjem se razporedijo po kompetencah in položaju ocenjevalcev, povratne informacije pa se združijo in pripravijo za pregled. Informacije, pridobi z ocenjevanjem, se neposredno predstavi ocenjevanim posameznikom, njihovim nadrejenim ali obojim. S pomočjo kadrovskega oddelka in specializiranih zunanjih svetovalcev (v našem primeru podjetje Pro Acta, d. o. o.),

se nato poskrbi za strokovno interpretacijo pridobljenih podatkov in ustrezno strokovno svetovanje ocenjevanim. [5]

## 2.1 Koristi in pasti uvajanja analize 360°

Razvoj novih zmožnosti posameznika je odvisen od uporabe sedanjega znanja in izkušenj. Analiza kompetenc 360° vam nudi podrobno povratno informacijo o sedanjih zmožnostih posameznika. Razvoj učinkovitega vedenja se začne z zavedanjem le-tega. Analiza kompetenc 360° pomaga posamezniku pri razumevanju njegovih prednosti in slabosti na različnih področjih ter mu nudi opis napredka na izbranih področjih. [5]

Analiza 360° ponuja tako posamezniku kot tudi podjetju številne koristi. Zaposleni večinoma želijo biti uspešni pri svojem delu. Mnogi se ne zavedajo vpliva svojega vedenja na učinkovitost opravljanja nalog pri delu ter na sodelavce. S pomočjo analize 360° zaposleni dobijo ključne informacije, ki jim pomagajo potrditi njihova predvidevanja glede svojih prednosti in jim pomagajo odkriti področja, na katerih se morajo izboljšati. Povratne informacije dobijo tako od nadrejenih, sodelavcev kot od podrejenih, kar jim omogoča kritično presojanje svojega zaznavanja z zaznavanjem drugih ljudi. S tem pa posamezniku pomagamo videti sebe skozi oči drugih. Uspešna uvedba analize 360° lahko pripomore k izboljšanim odnosom in komuniciranju med zaposlenimi in pomaga doseči in preseči pričakovanja, ki jih ima podjetje do njih. Prav tako pri posamezniku spodbudi osebni razvoj in osebno mojstrstvo, ki je lahko ključ do medosebne učinkovitosti. Spodbuja in izboljšuje posameznikovo ter skupinsko delovno uspešnost, zmanjšuje in odpravlja ovire, ki preprečujejo ustvarjalnost in izboljšuje komunikacijo na nivoju celotnega podjetja, še posebej med ljudmi vključenimi v analizo. [5]

Poleg številnih prednosti, ki jih ponuja ocenjevanje po metodi 360° pa lahko v primeru nestrokovne in neprimerne izvedbe analize naletimo na številne pasti, ki prinesejo več škode kot koristi. Ena izmed pasti je lahko neiskreno ocenjevanje. Zaposleni v strahu predi maščevanjem ali zamerami sodelavcev

ter nadrejenih v primeru slabih ocen ne podajo objektivne ocene. Nekateri lahko nekritično zavračajo povratne informacije ali so skeptični do le-teh in se sprašujejo, kako zanesljivi so na tak način pridobljeni podatki. Lahko prihaja do sporov in zamer med zaposlenimi. Bojijo se, da bi lahko zaradi slabših podanih ocen prišlo do slabih medsebojnih odnosov in izgube prijateljev ter kolegov. [12]

## 2.2 Kako deluje

Analiza se nanaša na izbrane kompetence, ki so pomembne za organizacijski ali vodstveni uspeh. Vprašalniki, ki so vključeni v analizo in jih zaposleni prejmejo, vključujejo kompetence, pomembne za podjetje. Posamezne kompetence so razdelane na določena ključna vedenja, ki naj bi jih oseba imela na določenem delovnem mestu. Spodaj navajam primer izbranih kompetenc in ključnih vedenj. [5]

### Globalno razmišljanje

- Prepozna vpliv globalizacije na naš posel.
- Kaže prilagodljivost, ki je potrebna za uspeh v globalnem okolju.
- Prizadeva si pridobiti raznolike izkušnje, ki so potrebne pri opravljanju posla.
- Pri svojih odločitvah upošteva globalizacijo.
- Drugim pomaga, da bolje razumejo vpliv globalizacije.

### Globalno razmišljanje

- Ustvarja in izraža jasno vizijo za podjetje.
- Učinkovito vključuje ljudi v odločanje.

- Ljudi navdušuje k uresničevanju vizije.
- Zna postaviti cilje.
- Jasno določi prioritete.

## 2.3 Ocenjevanje kompetenc

Vprašalniki, ki jih prejmejo zaposleni se ocenjujejo z vseh vidikov. Vsak izmed sodelujočih ocenjuje svoje nadrejene, sodelavce, podrejene ter se tudi samoocenjuje. Na vprašanja, ki niso odprtega tipa, sodelujoči odgovorijo z eno izmed ocen v spodnji tabeli [5].

Opis	Ocena
Neučinkovito: Kaže na neprimerno vedenje na tem področju ali pa sploh pomanjkanje tovrstnega vedenja.	1
Malo učinkovito: To vedenje ni izraženo konsistentno in ima malo učinka.	2
Učinkovito: Kaže konsistentno vedenje, ki je vedno učinkovito.	3
Zelo učinkovito: Kaže vedenje, ki je izjemno konsistentno in dosega izjemen uspeh.	4
Najbolj učinkovito: Med najboljšimi je to najboljše na tem področju. Vzor za druge.	5
To se ne nanaša na sodelavca.	0

Tabela 2.1: Ocene za vprašanja, ki niso odprtega tipa [5].

Analiza lahko vključuje več vprašanj odprtega tipa, ki omogočajo opisni odgovor. Primer vprašanj, na katera ocenjevalci odgovorijo z opisnim odgovorom [5]:

1. Katere so glavne prednosti te osebe?

- 
2. Kaj tej osebi predlagate, da naj spremeni, da bi bil/a bolj uspeš(e)n/a vodja?





## Poglavje 3

# Pregled programskih tehnologij in orodij za razvoj aplikacije

### 3.1 Razvojna orodja

Za delovanje in razvoj spletne aplikacije smo najprej s podporo za programski jezik PHP in relacijsko podatkovno bazo MySQL postavili spletni strežnik Apache. To smo storili z uporabo odprtokodnega programskega paketa XAMPP. Z orodjem Power Designer smo načrtovali in izdelali podatkovno bazo. Za administracijo podatkovne baze smo uporabili orodje phpMyAdmin. Programsko kodo, označevalni jezik HTML in kaskadne slogovne predloge CSS smo pisali in urejali v urejevalniku besedil Sublime Text 3. Za vodenje različic in shranjevanje zgodovine sprememb programske kode pa smo uporabili spletni repozitorij BitBucket. Spletno aplikacijo smo izdelali z uporabo ogrodja Laravel.

#### **XAMPP**

XAMPP je odprtokodni programski paket za operacijski sistem Windows, ki vsebuje Apache spletni strežnik, MariaDB podatkovno zbirko ter prevajalnike za programska jezika PHP ter Perl. Na voljo je tudi različica za operacijske sisteme Linux (LAMP) in Mac OS (MAMP), ki omogoča enostavno

poganjanje lokalnega spletnega strežnika Apache, lahko pa ga uporabimo tudi kot popolnoma delujoč spletni strežnik. V osnovi je namenjen izdelavi dinamičnih spletnih strani in aplikacij, ki so napisane v programskem jeziku PHP ali PEARL, ter za shranjevanje podatkov uporabljajo relacijsko podatkovno bazo SQL. Lahko ga uporabljamo tudi za razvoj spletnih strani in aplikacij, ki tečejo na drugih programskih jezikih. [15]

### **PowerDesigner**

Je orodje namenjeno načrtovanju in izdelavi podatkovnih baz ter grafičnemu načrtovanju relacijskih diagramov in modeliranju različnih tipov podatkovnih modelov. Poleg izdelave podatkovnih modelov omogoča tudi modeliranje poslovnih procesov, aplikativno modeliranje z uporabo UML diagramske tehnike, XML modeliranje in razne druge tehnike, s katerimi si lahko pomagamo pri razvoju informacijskih rešitev. [13]

### **Orodje phpMyAdmin**

Je programsko orodje napisano v jeziku PHP in je namenjeno administraciji MySQL in MariaDB podatkovne zbirke. Do orodja dostopamo preko spletnega brskalnika. Vsebuje grafični vmesnik preko katerega lahko enostavno dodajamo, urejamo in odstranjujemo posamezne tabele, polja, vrstice ali celotne podatkovne zbirke. Omogoča tudi urejanje in izvajanje SQL stavkov, upravljanje uporabnikov in njihovih pravic ter mnogo drugih naprednih funkcij, ki jih potrebujemo pri administraciji podatkovnih baz.

### **Urejevalnik besedila Sublime Text 3**

Sublime Text je napredni urejevalnik besedil in programskih kod. Podpira sintakse različnih programskih jezikov in ima močan nabor funkcij, kot so označevanje sintakse in drugih razpoznavnih elementov besedila glede na pomen vsebine. Odlikujejo ga hitrost, preprosta uporaba in razširljivost dodatnih možnosti urejevalnika z uporabo vtičnikov.

### **Program za vodenje različic BitBucket**

Je program za vodenje različic (angl. revision control system) in shranjevanje zgodovine sprememb v programski kodi. Namenjen je skupinskemu razvoju pisanja programske opreme. Omogoča vzporedno delo na več različicah (vejah) programske kode in skrbi za usklajenost različic vseh članov skupine. Uporabljamo ga lahko tudi za izdelovanje varnostnih kopij programske kode in drugih besedil.

### **Ogrodje Laravel**

Laravel je odprtokodno ogrodje za izdelovanje spletnih aplikacij in strani po arhitekturi MVC (Model-View-Controller). MVC arhitektura razdeli zgradbo aplikacije v tri plasti: Model, View (pogled) in Controller (krmilnik) [11]. Ogrodje je napisano v programskem jeziku PHP. Programerjem omogoča hiter in varen razvoj spletnih aplikacij. Vsebuje veliko naprednih komponent, ki močno olajšajo in pohitijo razvoj spletnih aplikacij. Vsebuje komponente za avtentikacijo in avtorizacijo uporabnikov, filtriranje http zahtevkov, preprosto povpraševanje podatkovne baze, pošiljanje elektronskih sporočil, obveščanje o dogodkih in napakah pri izvajanju aplikacije, komponente za delo z datotekami, odstranjevanje strani in tabel ter mnogo drugih komponent. Poleg vseh že vključenih komponent lahko dodajamo veliko dodatnih neodvisnih komponent, ki se nahajajo na za to namenjenih spletnih repozitorijih. Dodatne komponente se dodaja in posodablja z uporabo orodja Composer. [7]

### **Orodje Composer**

Composer je orodje za upravljanje paketov in odvisnosti za programski jezik PHP. Omogoča deklariranje in namestitve odvisnih knjižnic in paketov, ki jih uporabljamo pri razvoju aplikacije. Najprej vse pakete in odvisnosti deklarira v posebno datoteko (composer.json), nato pa z nekaj enostavnimi ukazi, ki jih vnesemo v ukazno vrstico, avtomatsko prenese in namesti vse deklarirane pakete ter odvisnosti v trenutni projekt. [1]

## 3.2 Opis uporabljenih tehnologij

### HTML

Kratika HTML (angl. Hyper Text Markup Language) je oznaka za označevalni jezik, ki se uporablja za izdelovanje spletnih strani. Je osnovna tehnologija za razvoj in prikaz spletnih strani. Vsebuje elemente (ukaze), ki jim rečemo tudi značke. S temi elementi označimo besedilo, ki brskalnikom pove, kako naj prikažejo dokument. Obstajata dve vrsti označevalnih elementov (značk):

- Samostojne značke (npr. `<br>` in `<img>`)
- Začetne in končne značke (npr. `<head></head>`, `<p></p>`)

Samostojne značke ne potrebujejo zaključka, začetne in končne značke pa pri vsaki uporabi potrebujejo končno značko. Končna značka je enaka začetni znački, razlikujeta se samo po dodatnem znaku (/), ki je postavljen pred imenom značke. Jezik omogoča elektronsko objavo dokumentov z besedili, fotografijami, tabelami, seznamami in obrazci ter preusmerjanje na druge dokumente s pomočjo hiperpovezav. [4]

### CSS

CSS (angl. Cascading Style Sheets) v slovenskem jeziku poznamo pod imenom kaskadne slogovne predloge, ki se uporabljajo za oblikovanje spletnih strani. Z njimi definiramo stil posameznih HTML značk na spletni strani. HTML elementom lahko določimo vrsto oblikovnih lastnosti, kot so barve, odmiki, velikosti, poravnave, obrobe ... HTML je namenjen logičnemu oblikovanju posameznih elementov na spletni strani, kar pomeni, da z njim samo določimo, kakšne vrste je posamezni element na strani (npr. odstavek, seznam, naslov, obrazec). Z uporabo CSS-ja določimo oblikovne lastnosti HTML značk. S tem ločimo logično in stilsko oblikovanje strani, kar poveča preglednost napisane kode in možnost uporabe iste slogovne predloge na več

straneh. [2]

## PHP

PHP (PHP: Hypertext Preprocessor) je razširjen odprtokodni strežniški programski jezik, ki se uporablja za izdelavo dinamičnih spletnih strani. Je skriptni jezik in spada v skupino prevajalnih jezikov, za katere je značilno, da se na začetku izvedbe ne prevaja celotna izvorna koda, temveč se prevaja/interpretira sproti. Jezik se običajno uporablja na spletnih strežnikih, ki imajo funkcijo, s katero interpretirano kodo pošljejo spletnim brskalnikom v obliki HTML kode. PHP kodo je potrebno pisati med značkami `<?php` in `?>`, saj le tako prevajalnik na strežniku ve, kateri del kode je potrebno prevesti. Zaradi preproste uporabe je PHP danes eden najbolj priljubljenih in razširjenih jezikov za izdelavo dinamičnih spletnih strani in aplikacij. [3]

## SQL

SQL (angl. Structured Query Language) ali strukturirani poizvedovalni jezik za delo s podatkovnimi bazami je najbolj razširjen in standardiziran povpraševalnik za delo s podatkovnimi bazami, kot so SQL server, Oracle, MySQL, Sybase in Access. SQL omogoči uporabniku dostop do zbirk podatkov ali podatkovnih baz. Poizvedovanje je najbolj pogosta operacija v SQL-u, izvrši pa se s SELECT stavkom. Poleg SELECT stavka so pomembni še ukazi za vstavljanje novih zapisov v podatkovno zbirko (INSERT), posodabljanje obstoječih zapisov (UPDATE) in odstranjevanje obstoječih zapisov. (DELETE). [14]

## JavaScript in jQuery

JavaScript je objektni skriptni programski jezik, ki tako kot PHP spada v skupino tolmačev. Uporablja se za dodajanje interaktivnosti spletnim stranem. Prevajalnik za JavaScript je danes vgrajen v večini novejših brskalnikih. Prevajalnik za JavaScript aktiviramo z vključitvijo `<script>` značke v HTML dokumentih. V HTML dokumentih lahko vključimo poljubno `<script>` značk,

ki se lahko nahajajo na poljubnih mestih, tako v glavi (`<head>`) kot v telesu (`<body>`). Z označevalnim jezikom HTML in stilskimi predlogami CSS tvori jedro za razvoj dinamičnih spletnih strani in aplikacij. V osnovi se JavaScript koda izvaja na strani odjemalca v spletnih brskalnikih, z razvojem ogrodij, kot je Node.js, pa se vse pogosteje uporablja za izvajanje operacij na spletnih strežnikih. [6] jQuery je knjižnica napisana v skriptnem jeziku JavaScript. Uporabimo jo tako, da v glavo ali nogo HTML dokumenta vstavimo značko `<script src='jquery.js'></script>`. Knjižnica vsebuje napredne funkcije, ki omogočajo lažjo izbiro in manipulacijo HTML značk, manipulacijo dogodkov html značk in strani, uporabo že pripravljenih efektov in animacij, asinhrono komunikacijo med odjemalcem in strežnikom ter različna druga orodja.

## **AJAX**

AJAX (asinhroni JavaScript in XML) je skupek spletnih tehnik, ki omogočajo asinhrono komuniciranje odjemalca s spletnim strežnikom brez potrebe po ponovnem nalaganju celotne spletne strani. Asinhrono komuniciranje odjemalca s strežnikom poteka s pomočjo JavaScript XMLHttpRequest objektov. Prednost asinhronne komunikacije je v tem, da v času ko poteka izmenjava podatkov med odjemalcem in strežnikom, uporabnik nemoteno lahko uporablja spletno stran. [10]

## Poglavje 4

# Opis funkcionalnosti in potek aplikacije

Spletna aplikacija, ki smo jo razvili za izvajanje ocenjevanja po metodi 360° ima dvouporabniško strukturo. Uporabnikom aplikacije lahko določimo vlogo skrbnika sistema (administratorja) ali vlogo navadnega uporabnika. Uporabniku, ki mu je dodana vloga skrbnika sistema, ima dostop do vseh razdelkov aplikacije. Preko zalednega sistema aplikacije vnaša in ureja vse potrebne parametre za izvajanje ocenjevanja ter spremlja in nadzira potek ocenjevanja. Ko so v aplikacijo vneseni vsi ključni parametri, ki so potrebni za izvedbo ocenjevanja, skrbnik sistema vsem vnesenim uporabnikom na njihove elektronske naslove razpošlje vabila k ocenjevanju, v katerih je poleg informacij o ocenjevanju tudi spletni naslov, preko katerega dostopajo do vprašalnikov.

### 4.1 Skrbnik sistema

#### 4.1.1 Prijava v aplikacijo in ponastavitev gesla

Skrbniku sistema se ob prihodu na spletni naslov, kjer je nameščena spletna aplikacija, najprej prikaže obrazec za prijavo v aplikacijo. Vsak uporabnik se v aplikacijo prijavi s svojim elektronskim naslovom in geslom.

The image shows a login interface for 'Analiza 360°'. At the top, the word 'Prijava' is centered. Below it are two input fields: 'E-mail' and 'Geslo'. A checkbox labeled 'Zapomni si me' is positioned below the password field. A 'Prijava' button is centered below the checkbox. At the bottom of the form area, there is a link: 'Ste pozabili geslo? Ponastavite geslo.'. The footer of the page displays 'Analiza 360°' and '©Pro Acta d.o.o.'.

Slika 4.1: Prijava uporabnika v aplikacijo.

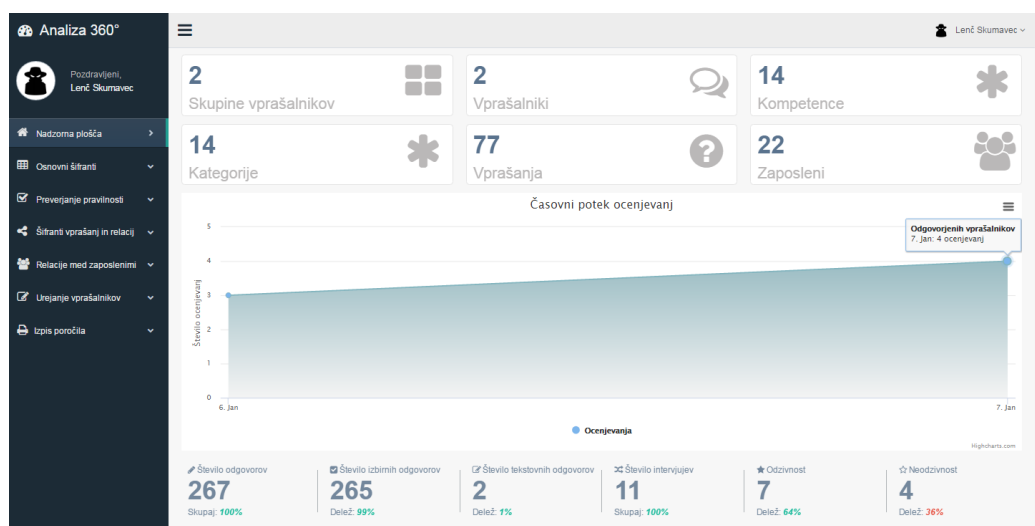
Vsak uporabnik lahko v primeru, da pozabi svoje geslo, le-tega ponovno ponastavi, kar stori s klikom na povezavo za ponastavitev osebnega gesla, ki ga pripelje do obrazca za ponastavitev gesla. V obrazec vnese svoj elektronski naslov. Aplikacija preveri, če je uporabnik s tem elektronskim naslovom registriran v sistemu. V primeru, da je vneseni elektronski naslov veljaven in zapisan v sistem, uporabnik na ta elektronski naslov prejme navodila za ponastavitev svojega gesla in povezavo, ki vsebuje varnostni žeton ter ga pripelje do novega obrazca, preko katerega nato nastavi svoje geslo za vstop.

#### 4.1.2 Nadzorna plošča

Po uspešni prijavi v spletno aplikacijo uporabnik prispe na nadzorno ploščo. V tem razdelku ima administrator sistema pregled nad celotnim potekom ocenjevanja. Uporabniku se na levi strani zaslona prikaže glavni stranski meni, ki vsebuje povezave do vseh razdelkov spletne aplikacije. Glavni meni vsebuje sedem večjih razdelkov, v katere vnašamo in urejamo vse potrebne parametre za izvedbo ocenjevanja. Na dnu menija sta na voljo gumba za



izvoz podatkov in odjavo iz sistema. Na desni strani zaslona so najprej prikazani podatki o vnesenih parametrih, ki so pomembni za izvedbo ocenjevanja, nato sledi grafični prikaz poteka ocenjevanja. Iz grafa skrbnik sistema lahko razbere datum prvega in zadnjega ocenjevanja ter število ocenjevanj na posamezni datum. Pod grafičnim prikazom je razdelek, kjer lahko uporabnik vidi, koliko je bilo do tega trenutka odgovorjenih vprašanj, kolikšno je število izbirnih in opisnih odgovorov, koliko je vseh ocenjevanj, koliko ljudi je že odgovarjalo na vprašalnike in koliko jih še ni.



Slika 4.2: Nadzorna plošča skrbnika spletne aplikacije.

Razdelek nižje so v podatkovni tabeli prikazani vsi uporabniki, ki sodelujejo pri ocenjevanju. Za vsakega uporabnika je prikazano število prejetih vabil na ocenjevanje ter število že ocenjenih vabil, v zadnjem stolpcu pa je gumb za pošiljanje vabil. S pritiskom na gumb, uporabnik na elektronski naslov prejme vabila za ocenjevanja, a le še neizpolnjena. V primeru, da oseba ne sodeluje pri ocenjevanju ali pa je na poslana vabila že odgovarjala, aplikacija elektronskega sporočila ne pošlje.

### 4.1.3 Vnos in urejanje šifrantov

V tem razdelku preko obrazcev vnašamo in urejamo šifrante, ki so potrebni za izvedbo ocenjevanja. Vsi vneseni šifranti, kot so skupine vprašalnikov, vprašalniki, kompetence, kategorije, vprašanja in uporabniki, so prikazani v podatkovnih tabelah.

Pričnemo z vnašanjem skupin vprašalnikov. Sledi vnos posameznih vprašalnikov, ki mu sledi vnos kompetenc. Za vsako izmed vnesenih kompetenc določimo kateremu vprašalniku pripada. V naslednjem koraku dodajamo kategorije. Vsaka izmed dodanih kategorij pripada eni izmed prej vnesenih kompetenc. Sledi vnos posameznih vprašanj, ki jim določimo kateri kategoriji pripadajo. V zadnjem koraku vnašamo še uporabnike, ki bodo sodelovali pri ocenjevanju.

Skupine vprašalnikov

Zaporedna številka:

2

Opis:

Vprašalniki za sodelavce

Opomba:

Dodaj

Shrani

Uredi

Izbrisi

Skupine vprašalnikov

	Zaporedna številka ↕	Opis	Opomba
<input type="checkbox"/>	1	Vprašalniki za vodje	
<input checked="" type="checkbox"/>	2	Vprašalniki za sodelavce	

Stran 1 od 1

10

Prikazanih 1 - 2 od 2

Slika 4.3: Prikaz obrazca in podatkovne tabele za vnos in urejanje skupin vprašalnikov.

#### 4.1.4 Preverjanje pravilnosti vnesenih elektronskih naslovov

Po vnosu vseh potrebnih šifrantov in uporabnikov sledi preverjanje pravilnosti vnesenih elektronskih naslovov uporabnikov, na katere se pošljejo elektronska sporočila, ki vsebujejo navodila in povezave do spletnih vprašalnikov, zato je pomembno, da je vsak izmed vnešenih elektronskih naslovov veljaven. Uporabniki in kriteriji, ki jih uporabimo za preverjanje elektronskih naslovov, so prikazani v podatkovni tabeli.



#	Elektronski naslov	Veljavnost zapisa	SMTP	MX-zapis
5	lenc@conforma.si	✓	✓	✓
6	killerade@gmail.com	✓	✓	✓
7	asdfkj@jkkasdf	✗	✗	✗
8	frank@sobotka.si	✓	✗	✗
9	zigg@sobotka.si	✓	✗	✗
10	jimmy@mcnulty.si	✓	✗	✗
11	rob@xn--nejdr-udb.si	✓	✗	✗

Slika 4.4: Prikaz preverjanja veljavnosti elektronskih naslovov.

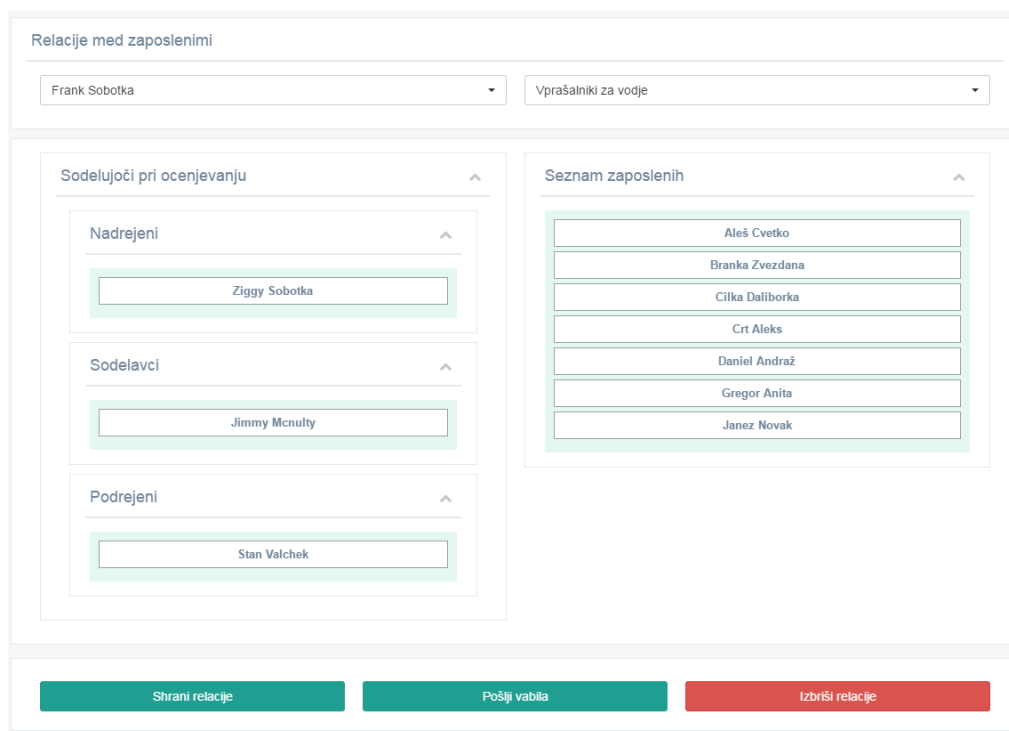
#### 4.1.5 Določanje relacij med uporabniki in vnos spremnega besedila

Preden lahko pričnemo z ocenjevanjem moramo določiti relacije med uporabniki in vnesti spremno besedilo. Posameznemu uporabniku določimo, kdo vse ga bo ocenjeval, v kakšni relaciji so z njim ocenjevalci ter skupino vprašalnikov, na katero bodo odgovarjali. V razdelku za določanje relacij med uporabniki sta dva spustna seznama in neoštevilčeni seznam, v katere poljubno dodajamo uporabnike.

V prvem spustnem seznamu izberemo zaposlenega in mu določimo skupino

vprašalnikov, na katere bodo odgovarjali vsi sodelujoči pri ocenjevanju. Skupino vprašalnikov izberemo v drugem spustnem seznamu, kjer imamo na voljo seznam vseh skupin vprašalnikov, ki vsebujejo dodeljene vprašalnike. Izbranemu zaposlenemu določimo, kdo ga bo ocenjeval in v kakšni relaciji je zaposleni z izbranimi ocenjevalci. To storimo tako, da uporabnike s seznama vseh uporabnikov na način "drag and drop" razporedimo v sezname sodelujočih.

Ko vse zaposlene ustrezno razporedimo po seznamih, sledi vnos spremnega besedila. To je besedilo, ki se pošlje vsem izbranim ocenjevalcem na njihove elektronske naslove pri pošiljanju vabil za ocenjevanje. V grafični urejevalnik besedila vnesemo poljubno besedilo in ga po potrebi ustrezno oblikujemo. Poleg besedila lahko vnašamo tudi vnaprej določene značke, ki se pri pošiljanju vabil na elektronske naslove zamenjajo z ustreznimi vrednostmi.



Relacije med zaposlenimi

Frank Sobotka Vprašalniki za vodje

Sodelujoči pri ocenjevanju

Nadrejeni

Ziggy Sobotka

Sodelavci

Jimmy McNulty

Podrejeni

Stan Valchek

Seznam zaposlenih

Aleš Cvetko

Branka Zvezdana

Čilka Daliborka

Crt Aleks

Daniel Andraž

Gregor Anita

Janez Novak

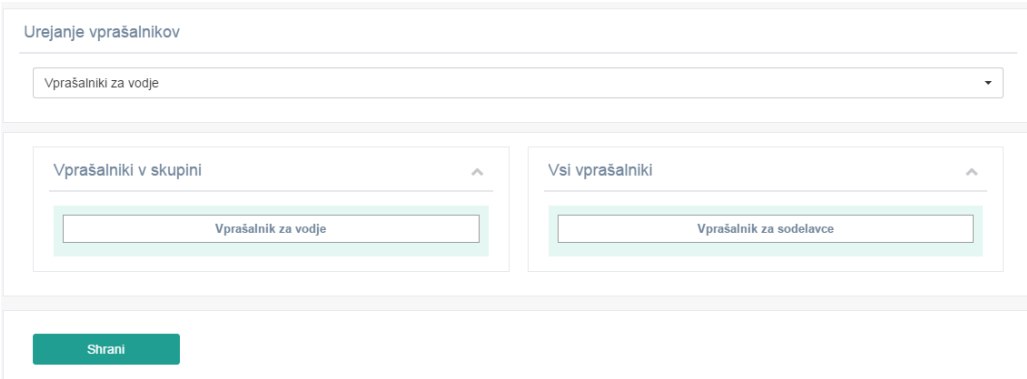
Shrani relacije Pošlji vabila Izbrši relacije

Slika 4.5: Prikaz urejanja relacij med uporabniki.

### 4.1.6 Urejanje vprašalnikov

V zadnjem koraku skupinam vprašalnikov dodelimo ustrezne vprašalnike. S spustnega seznama najprej izberemo skupino vprašalnikov. Nato se nam v spodnjem desnem seznamu pojavi seznam vseh vprašalnikov, ki smo jih vnesli pri dodajanju šifrantov. Z uporabo načina "drag and drop" s seznama vseh vprašalnikov izbrane vprašalnike prenesemo v seznam na levi strani.

Vsak vprašalnik lahko določimo za več skupin vprašalnikov. Tako imamo lahko določeno skupino vprašanj v več skupinah vprašalnikov. Vrstni red dodajanja vprašalnikov je pomemben. V takšnem vrstnem redu kot razporedimo vprašalnike iz desnega seznama v levi seznam bodo vprašanja prikazana uporabnikom pri ocenjevanju.

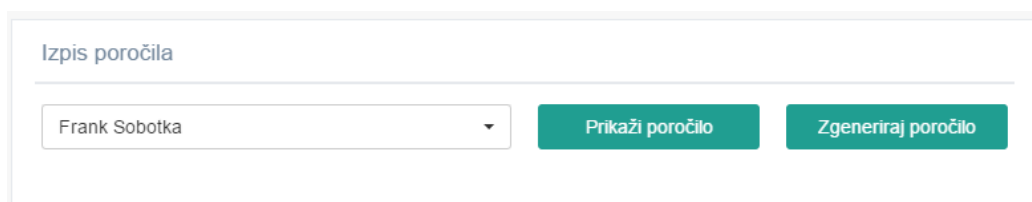


Slika 4.6: Prikaz razdelka za urejanje skupin vprašalnikov.

### 4.1.7 Izpis poročil

Po končanem ocenjevanju sledi izdelava poročil za zaposlene. V zadnjem zavihku stranskega menija imamo razdelek **Izpis poročil**. S spustnega seznama najprej izberemo posameznega uporabnika, za katerega želimo izdelati končno poročilo. S pritiskom na gumb **Prikaži poročilo** aplikacija ustvari izčrpno poročilo za izbranega zaposlenega. Poročilo uporabnika se odpre v

novem zavihku spletnega brskalnika. Ustvarjeno poročilo lahko pretvorimo v PDF zapis in ga poljubno shranimo na svoj računalnik. To storimo tako, da v spletnem brskalniku izberemo opcijo **Natisni**, ki nam v novem oknu prikaže poročilo v obliki za tisk. Z izbiro opcije **Shrani kot PDF** izpisano poročilo v PDF zapisu shranimo na poljuben medij.



Izpis poročila

Frank Sobotka ▼

Prikaži poročilo

Zgeneriraj poročilo

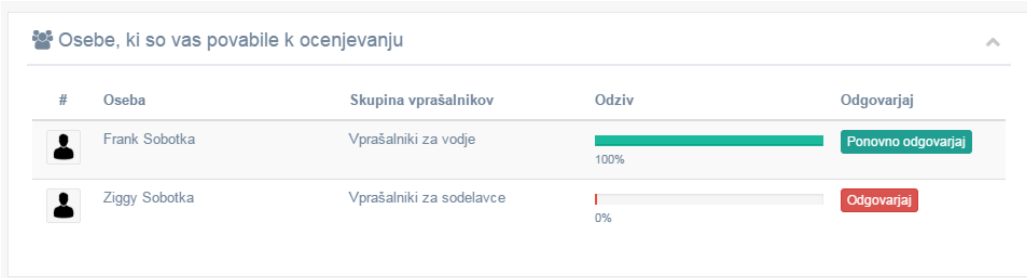
Slika 4.7: Prikaz obrazca za izpis poročila za posameznega uporabnika.



## 4.2 Uporabnik

### 4.2.1 Prijava v aplikacijo in ponastavitev gesla

Tako kot v primeru skrbnika sistema se tudi uporabniki ob prihodu na spletni naslov kjer se nahaja spletna aplikacija morajo najprej prijaviti v aplikacijo. V aplikacijo se prijavimo z vnosom gesla in elektronskega naslova, ki ga skrbnik sistema vnese pri vnašanju uporabnikov. V primeru, da gre za prvo prijavo v sistem, si moramo najprej nastaviti naše geslo. Postopek ponastavljanja gesla je enak kot pri skrbniku sistema.

Po uspešni prijavi prispemo na začetno stran, kjer se nam na zaslonu izpiše seznam vseh vabil na ocenjevanje. V prvem stolpcu je zapisana oseba, ki nas je povabila na ocenjevanje. V naslednji koloni je naziv skupine vprašalnikov, ki jih bomo ocenjevali. V tretji koloni je podatek o tem, ali smo že odgovarjali na poslano vabilo, zadnja pa vsebuje gumb s povezavo do vprašalnikov za osebo, ki nas je povabila k ocenjevanju.



#	Oseba	Skupina vprašalnikov	Odziv	Odgovarjaj
	Frank Sobotka	Vprašalniki za vodje	<div><div></div></div> 100%	<a href="#">Ponovno odgovarjaj</a>
	Ziggy Sobotka	Vprašalniki za sodelavce	<div><div></div></div> 0%	<a href="#">Odgovarjaj</a>

Slika 4.8: Prikaz prejetih vabil za ocenjevanje.

#### 4.2.2 Ocenjevanje vprašalnikov

Do vprašalnikov pridemo preko spletnih povezav, ki smo jih v vabilu za ocenjevanje prejeli na elektronski naslov. Na vrhu zaslona imamo podatek o tem, za koga odgovarjamo na zastavljena vprašanja, in podatke o skupini vprašalnikov, kompetencah in kategorijah, ki jim pripadajo prikazana vprašanja.

Vprašanja so lahko izbirnega ali opisnega tipa. Na vprašanja izbirnega tipa odgovorimo z izbiro enega izmed vnaprej ponujenih odgovorov. Na vprašanja opisnega tipa odgovorimo z vnosom besedila v tekstovno polje. Odgovoriti je potrebno na vsa vprašanja, v nasprotnem primeru nas aplikacija opozori, da je za nadaljevanje potrebno odgovoriti na vsa vprašanja. S klikom na gumb Naprej se na zaslonu izpiše nov nabor vprašanj, ki jih vsebuje vprašalnik. Ko odgovorimo na vsa zastavljena vprašanja, se na zaslonu izpiše obvestilo o zaključku ocenjevanja. S klikom na gumb Zaključí se podani odgovori zapišejo v podatkovno bazo. V primeru, da želimo ponovno odgovarjati na zastavljena vprašanja, kliknemo gumb Ponovno odgovarjaj.

Ko odgovorimo na vsa zastavljena vprašanja se nam na zaslonu izpiše obvestilo o zaključku ocenjevanja. S klikom na gumb Zaključí se podani odgovori zapišejo v podatkovno bazo. V primeru, da želimo ponovno odgovarjati na zastavljena vprašanja kliknemo na gumb Ponovno odgovarjaj.

Vprašalnik izpolnjujete za osebo Ziggy Sobotka

Vprašalnik: Vprašalnik za sodelavce  
Kompetenca: Upravljanje s spremembami  
Kategorija: Upravljanje s spremembami

Pri drugih vzpodbuja ustvarjalnost in inovativnost

☐ Nikoli ☐ Redko ☐ Pogosto ☒ Vedno ☐ Ne morem odgovoriti

Učinkovito spremeni ustvarjalne ideje v poslovne rezultate

☐ Nikoli ☒ Redko ☐ Pogosto ☐ Vedno ☐ Ne morem odgovoriti

Katere so vaše prednosti? (prosimo, napišite vsaj tri)

Lorem ipsum is simply dummy text of the printing and typesetting industry. Lorem ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

Naprej

Slika 4.9: Prikaz ocenjevanja vprašalnika.

Vprašalnik izpolnjujete za osebo Ziggy Sobotka

Hvala za vaš čas in sodelovanje pri vprašalniku 360.

Ponovno odgovarjaj? Zaključ

Slika 4.10: Prikaz zaključka ocenjevanja.



## Poglavje 5

# Potek in razvoj ključnih rešitev

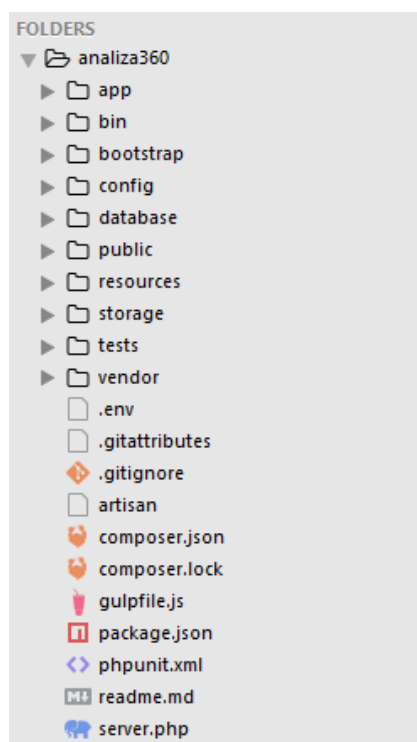
### 5.1 Namestitev in struktura ogrodja Laravel

Spletno aplikacijo smo izdelali s pomočjo odprtokodnega ogrodja Laravel verzije 5.2. ter ga namestili s pomočjo orodja Composer. V ukazni vrstici smo ustvarili nov projekt z naslovom analiza360, kar smo storili s spodnjim ukazom:

```
composer create-project --prefer-dist laravel/laravel analiza360
```

S tem ukazom je Composer na našem računalniku ustvaril novo mapo z nazivom analiza360 in vanjo namestil ogrodje. Poleg namestitve orodje poskrbi tudi za osnovno nastavitev ogrodja. S tem je bila namestitev ogrodja z vsemi pripadajočimi komponentami in odvisnostmi končana. V nastavitveni datoteki `.env` je bilo potrebno nastaviti še parametre za povezavo ogrodja s podatkovno bazo. Po končanem nastavljanju datoteke `.env` je bilo ogrodje pripravljeno za uporabo.

Struktura ogrodja je zasnovana tako, da predstavlja odlično izhodiščno točko razvijalcem tako za velike kot tudi manjše spletne aplikacije. Strukturo lahko poljubno prilagajamo svojim potrebam. Pri spreminjanju strukture je potrebno paziti le na to, da ne spreminjamo strukture map in datoteke, ki so



Slika 5.1: Struktura ogrodja Laravel.

ključne za delovanje ogrodja. Ključne datoteke in mape, ki smo jih potrebovali pri razvoju aplikacije so:

- app - v mapi so shranjeni vsi modeli, ki smo jih ustvarili za potrebe naše aplikacije in vsebujejo metode za manipulacijo s podatkovno bazo. Poleg modelov mapa vsebuje še vse krmilnike, ki vsebujejo povezovalno kodo med modeli in pogledi. Mapa vsebuje tudi datoteko `routes.php`, v katero smo zapisovali vse potrebne poti za delovanje naše aplikacije, ustvarjene vmesne mehanizme in nekatere ostale mape z vsebovanimi datotekami, ki so del jedra ogrodja.
- config - v mapi se nahajajo vse nastavitvene datoteke ogrodja. Vsaka datoteka vsebuje več tabel, v katerih s spremembo vrednosti določenega polja spreminjamo nastavitve posameznega dela ali komponente ogrodja.
- database - v mapi se nahajajo vsi migracijski razredi, ki smo jih ustvarili

za kreiranje tabel v podatkovni bazi.

- `public` - v tej mapi se nahajajo kaskadne stilske predloge CSS in vse JavaScript knjižnice, ki smo jih uporabili za razvoj funkcionalnosti na strani odjemalca. Poleg JavaScript knjižnic in kaskadnih stilskih predlog se v mapi nahajajo še vse ikone in ostali grafični material, ki smo jih uporabili pri izdelavi uporabniškega vmesnika aplikacije.
- `resources` - vsebuje poglede za registracijo in prijavo uporabnikov, ponastavitev gesel uporabnikov, izdelavo poročil in uporabniškega vmesnika aplikacije.
- `storage` - v mapi se nahaja dnevniška datoteka `.log`, kamor se zapisujejo vse napake in sledi, ki se zgodijo pri uporabi aplikacije.
- `.env` - je konfiguracijska datoteka, v kateri smo nastavili parametre za povezavo s podatkovno bazo in pošiljanje elektronskih sporočil. V datoteki se nahajajo še druge nastavitve, kot so nastavitve za način predpomnjenja in lokalizacija.
- `composer.json` - je konfiguracijska datoteka za orodje Composer, v kateri so zapisane vse odvisnosti in paketi za programski jezik PHP, ki smo jih uporabili pri razvoju aplikacije.

## 5.2 Načrtovanje in podatkovne baze

Razvoj spletne aplikacije smo pričeli z načrtovanjem podatkovnega modela. Podatkovni model smo načrtovali s pomočjo orodja Power Designer. Najprej smo izdelali konceptualni model podatkovne baze. Določili smo glavne entitete, ki so potrebne za izdelavo spletne aplikacije, ter vsaki določili potrebne attribute in tip atributov. Ko smo ustvarili vse potrebne entitete in attribute, smo določili še razmerja oziroma povezave med posameznimi entitetami ter kardinalnosti razmerij. Končan podatkovni model vsebuje štirinajst entitet,

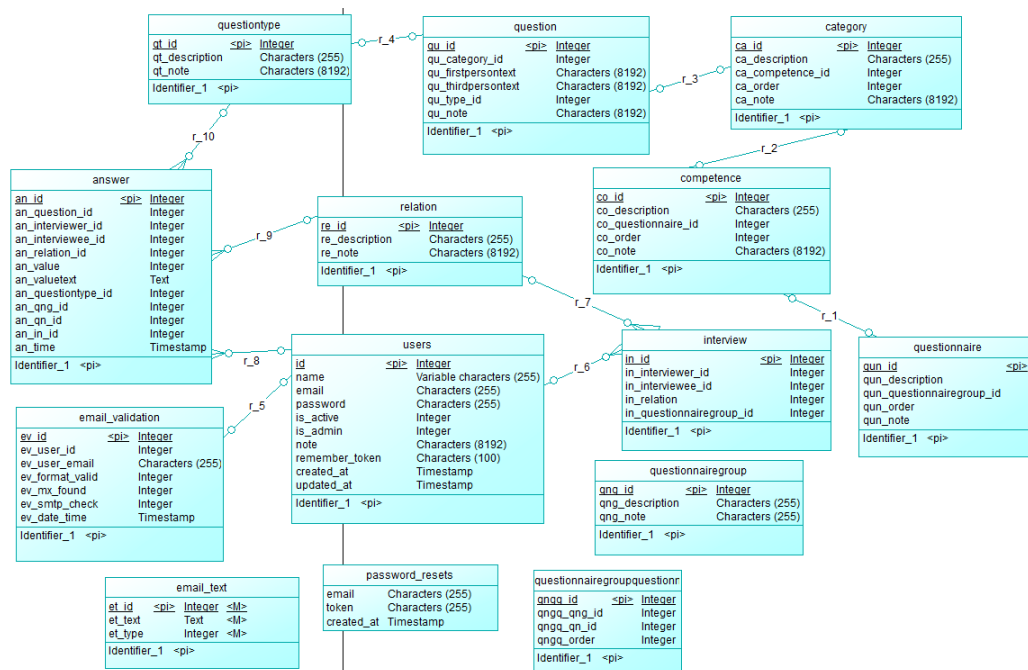
ki so potrebne za delovanje spletne aplikacije. Ko smo imeli končan konceptualni model z vsemi entitetami in določenimi povezavami med entitetami, smo opravili še postopek normalizacije.

Konceptualni model vsebuje naslednje entitete:

- questionnairegroup - entiteta, ki predstavlja skupino vprašalnikov. Vsebuje naziv skupine vprašalnikov in opombo.
- questionnaire - entiteta, ki predstavlja vprašalnike, ki jih določimo posamezni skupini vprašalnikov. Vsebuje naziv vprašalnika in opombo.
- questionnairegroupquestionnaire - entiteta vsebuje zapis o tem, kateri skupini vprašalnikov pripada določeni vprašalnik. Vsak vprašalnik lahko pripada več skupinam vprašalnikov. Vsebuje šifro skupine vprašalnikov, šifro posameznega vprašalnika, ki ga dodelimo izbrani skupini vprašalnikov, in vrstni red.
- competence - vmesna entiteta, ki predstavlja posamezno kompetenco in je povezana z entiteto questionnaire. Vsaka kompetenca lahko pripada le enemu vprašalniku, vsak vprašalnik pa lahko vsebuje več kompetenc. Vsebuje naziv kompetence, šifro vprašalnika, ki mu pripada, vrstni red in opombo.
- category - vmesna entiteta, ki predstavlja posamezno kategorijo znotraj določene kompetence. Povezana je z entiteto competence. Vsaka kategorija lahko pripada natanko eni kompetenci. Vsebuje naziv kategorije, šifro kompetence kateri pripada, vrstni red in opombo.
- question - entiteta, ki predstavlja posamezno vprašanje in spada v določeno kategorijo. Entiteta je povezana z entiteto category. Vsako vprašanje lahko pripada natanko eni kategoriji. Vsebuje šifro kategorije kateri pripada, vprašanje v prvi in tretji osebi ednine, šifro za tip vprašanja in opombo.

- `questiontype` - entiteta, ki predstavlja tip oziroma obliko vprašanja (izbirni tip, opisni tip). Povezana je z entitetama `question` in `answer`. Vsako vprašanje in odgovor sta lahko le enega tipa. Vsebuje opis tipa in opombo.
- `answer` - entiteta, ki predstavlja posamezni odgovor na zastavljeno vprašanje. Povezana je z entitetama `users` in `relation`. Vsak odgovor pripada natanko eni osebi, s katero je v eni izmed določenih relacij. Vsebuje šifro vprašanja, za katerega je podan odgovor, šifro ocenjevalne osebe, šifro ocenjevalca, šifro relacije, v kateri sta ocenjevana oseba in ocenjevalec, izbirni odgovor v primeru, da gre za odgovor na izbirni tip vprašanja, opisni odgovor, če gre za odgovor na opisni tip vprašanja, šifro tipa vprašanja, ki mu pripada, šifro skupine vprašalnikov, ki jim pripada, šifro vprašalnika, šifro intervjuja in zapis datuma ter časa podanega odgovora.
- `relation` - entiteta, ki predstavlja relacijo, v kateri sta ocenjevalec in uporabnik, ki povabi k ocenjevanju. Vsebuje opis relacije in opombo.
- `users` - entiteta, ki predstavlja posameznega uporabnika. Povezana je z entitetami `answer`, `interview` in `email_validation`. Vsebuje ime in priimek osebe, elektronski naslov, geslo, opombo, zapis o dodeljeni vlogi uporabnika, zapis o tem, ali ima uporabnik omogočeno prijavo v aplikacijo, opombo, datum vnosa uporabnika in datum posodobitve podatkov uporabnika.
- `interview` - entiteta, ki vsebuje šifro ocenjevalne osebe, šifro ocenjevalca, šifro relacije, v kateri sta ocenjevalec in ocenjevana oseba ter šifro skupine vprašalnikov, na katere ocenjevana oseba povabi ocenjevalca. Entiteta je povezana z entitetama `relation` in `users`.
- `password_reset` - entiteta, ki vsebuje podatke, ki so potrebni za ponastavljanje uporabniških gesel za prijavo v spletno aplikacijo.

- email.text - entiteta, ki vsebuje besedilo, ki ga uporabniki prejmejo na elektronske naslove in vsebuje navodila za dostop in ocenjevanje vprašalnikov.



Slika 5.2: Konceptualni model podatkovne baze.

Podatkovni model, ki smo ga napravili z orodjem Power Designer, smo nato implementirali v podatkovno bazo s komponento za ukazno vrstico Artisan. Najprej smo za vsako entiteto kreirali migracijski razred. Primer kreiranja migracijskega razreda za entiteto `users` v ukazni vrstici:

```
php artisan make:migration create_users_table
```

S tem ukazom ustvarimo migracijski razred za entiteto `users`. Ustvarjen razred vsebuje dve metodi: metodo `up` in metodo `down`. Metoda `up` je uporab-

ljena za kreiranje tabele. V njej definiramo ime tabele in vse stolpce ter tipe stolpcev, ki jih vsebuje tabela. Metoda `down` se izvede pri brisanju tabele. Za kreiranje tabel v podatkovni bazi, ki smo jih določili z migracijskimi razredi, moramo izvesti še ukaz:

```
php artisan migrate
```

V primeru, da se migracije ne izvedejo in pri kreiranju tabel v podatkovni bazi pride do napak, lahko bazo podatkov povrnemo v prvotno stanje z ukazom:

```
php artisan migrate:rollback
```

## 5.3 Registracija in prijava uporabnikov

Naslednji korak pri razvoju aplikacije je bil razvoj sistema za registracijo in prijavo uporabnikov. Laravel že v osnovi vsebuje komponento za registracijo in prijavo uporabnikov. Komponento namestimo z uporabo orodja Artisan. V ukazno vrstico najprej vnesemo naslednji ukaz:

```
php artisan make:auth
```

S tem ukazom se v datoteki `routes.php` najprej doda zapis `Route::auth()`. Metoda `auth` vsebuje vse potrebne poti za registracijo in prijavo uporabnikov ter ponastavljanje njihovih gesel.

Z izvedbo ukaza `php artisan make:auth` smo ustvarili osnovne poglede (angl. views) za registracijo in prijavo uporabnikov ter krmilnik (angl. controller), ki vsebuje logiko za registracijo in prijavo uporabnikov. Poleg pogledov za registracijo in prijavo uporabnikov smo ustvarili tudi poglede, ki se uporabijo pri pošiljanju elektronskih sporočil uporabnikom pri ponastavljanju.

```
367     public function auth()  
368     {  
369         // Authentication Routes...  
370         $this->get('login', 'Auth\AuthController@showLoginForm');  
371         $this->post('login', 'Auth\AuthController@login');  
372         $this->get('logout', 'Auth\AuthController@logout');  
373  
374         // Registration Routes...  
375         $this->get('register', 'Auth\AuthController@showRegistrationForm');  
376         $this->post('register', 'Auth\AuthController@register');  
377  
378         // Password Reset Routes...  
379         $this->get('password/reset/{token?}', 'Auth>PasswordController@showResetForm');  
380         $this->post('password/email', 'Auth>PasswordController@sendResetLinkEmail');  
381         $this->post('password/reset', 'Auth>PasswordController@reset');  
382     }  
383
```

Slika 5.3: Prikaz metode auth.

ju uporabniških gesel.

Logiko za registracijo in prijavo uporabnikov je bilo potrebno prilagoditi zahtevam naše aplikacije. Najprej je bilo potrebno prilagoditi logiko za registracijo uporabnikov. Preprečiti smo želeli, da bi se v aplikacijo lahko registriral vsak, ki pozna spletni naslov (angl. url) do obrazca za registracijo uporabnikov. Nove uporabnike v aplikacijo lahko dodaja izključno uporabnik z vlogo administratorja, zato je bilo potrebno poti, ki vodijo do obrazca za registracijo uporabnikov, zaščititi tako, da do obrazca za registracijo uporabnikov lahko dostopa le prijavljen uporabnik z dodano vlogo administratorja. Nato smo prilagodili še logiko za prijavo registriranih uporabnikov v sistem.

V osnovi se uporabnik v aplikacijo prijavi s svojim elektronskim naslovom in geslom. Ker pa ima administrator na voljo, da posameznemu uporabniku onemogoči prijavo, je bilo potrebno prilagoditi še metodo `login`, ki se nahaja v razredu `AuthenticatesUsers.php`. Metodo je bilo potrebno prilagoditi tako, da poleg vnesenega elektronskega naslova in gesla uporabnika v tabeli `users` preveri še polje `is_active`. Vrednost v polju `is_active` določa, ali je uporabniku omogočena prijava v aplikacijo.



## 5.4 Avtorizacija uporabnikov

Uporabnikom je lahko dodeljena vloga administratorja ali navadnega uporabnika. Po uspešni prijavi aplikacija uporabniku omeji dostop le do tistih razdelkov aplikacije, ki so dodeljeni določeni vlogi. Vloga uporabnika je določena v podatkovni tabeli `users` z zapisi vrednosti v polju `is_admin`. Če je vrednost polja `is_admin` ena, potem je uporabniku dodana vloga administratorja. V primeru, da je vrednost polja nič, je uporabniku dodeljena vloga navadnega uporabnika. Uporabnike z vlogo administratorja se v aplikacijo doda preko obrazca za registracijo uporabnikov. Vsem uporabnikom, ki so dodani preko obrazca za registracijo uporabnikov, se samodejno dodeli vloga administratorja. Administrator nato vnaša navadne uporabnike preko razdelka `Zaposleni` v zalednem sistemu aplikacije.

Avtorizacijo uporabnikov smo naredili z uporabo vmesnih mehanizmov (angl. *middleware*), ki jih vsebuje ogrodje Laravel. Vmesni mehanizem deluje tako, da filtrira vse http zahteve, ki vstopajo v aplikacijo. Vmesne mehanizme določimo po posamezni poti, skupini poti v datoteki `routes.php` ali pa jih dodelimo znotraj posameznih krmilnikov. Za vzpostavitev avtorizacije, ki ustreza zahtevam naše aplikacije, smo ustvarili naslednje vmesne mehanizme:

- `UserRole.php` - preveri če je uporabnik, ki je sprožil http zahtevek, prijavljen v aplikacijo in katera vloga mu je dodeljena. Če uporabnik ni prijavljen v sistem, ga vmesni mehanizem preusmeri na poljuben pogled.
- `Interview.php` - poleg preverjanja, če je uporabnik prijavljen v aplikacijo, preveri še, če ima uporabnik pravico dostopati do posameznega vprašalnika, ki se nahaja na spletnem naslovu, kateremu je dodeljen vmesni mehanizem.

Vmesni mehanizem ustvarimo z orodjem Artisan. Primer ukaza, s katerim smo ustvarili vmesni mehanizem za filtriranje http zahtevkov glede na vlogo

prijavljenega uporabnika:

```
php artisan make:middleware UserRole
```

## 5.5 Vnos in urejanje šifrantov

Vnos in urejanje šifrantov poteka preko obrazcev in podatkovnih tabel, kamor se shranjujejo vneseni podatki. Vnesene podatke urejamo tako, da izberemo ustrezno vrstico v podatkovni tabeli, nato pa se nam v polja obrazca vnesejo vrednosti za izbrano vrstico. Stare vrednosti lahko zamenjamo z novimi in jih shranimo v podatkovno bazo. Najprej smo izdelali krmilnik, ki vsebuje vso logiko za črpanje in shranjevanje podatkov za posamezni šifrant. Nato smo dodali še vse potrebne poti in pričeli z izdelavo pogledov.

Poglede smo izdelali s pomočjo jezika **Blade**, ki ga vsebuje Laravel. To smo storili tako, da smo nazivom ustvarjenih pogledov dodali končnico `blade.php`. Izdelali smo osnovni pogled, ki služi kot predloga, ki jo nato podedujejo ostali pogledi. S funkcijo `@yield` smo v osnovni predlogi določili mesto, kamor se gnezdi vsebina pogledov, ki podedujejo ta pogled. S tem smo se izognili potrebi po vnovičnem vključevanju kaskadnih stilskih predlog, JavaScript knjižnicam in html oznakam, ki se ponavljajo v vsakem na novo ustvarjenem pogledu. Poglede smo v osnovni pogled vdelali z uporabo funkcij `@extends` in `@section`.

Obrazce za vnos in urejanje podatkov smo naredili z uporabo HTML značk za izdelavo obrazcev. Podatkovno tabelo kamor se vnašajo vneseni podatki, smo izdelali v jeziku JavaScript. Pod obrazec za vnos in urejanje podatkov smo dodali še gumbе za dodajanje, shranjevanje, urejanje in brisanje podatkov.

## 5.6 Preverjanje pravilnosti elektronskih naslovov

Elektronske naslove uporabnikov aplikacije preverjamo z uporabo spletne storitve **Mailboxplayer** [8], ki je spletni API za preverjanje veljavnosti elektronskih naslovov. Za uporabo API-ja smo si na njihovi spletni strani najprej ustvarili uporabniški račun in s tem pridobili ključ, ki ga potrebujemo za dostop do API-ja.

Poleg dodanih poti in ustvarjenih pogledov za prikaz podatkov o pravilnosti elektronskih naslovov smo izdelali še krmilnik, ki vsebuje logiko za uporabo storitve **Mailboxplayer**. V krmilniku smo ustvarili metodo **email\_validation**, ki skrbi za preverjanje elektronskih naslovov vseh registriranih uporabnikov. Metoda najprej iz podatkovne baze pridobi uporabnike in njihove elektronske naslove. Nato pa za vsakega uporabnika naredimo http zahtevek na končno točko (angl. endpoint) **Mailboxplayer** storitve. Http zahtevek na končno točko API storitve naredimo z uporabo funkcij PHP knjižnice **cURL**. Vsak http zahtevek, ki ga pošljemo na API mora vsebovati tudi ključ in elektronski naslov, ki ga želimo preveriti. To storimo tako, da v spletni naslov pri http zahtevkih dodamo parametra **access\_key** in **elektronski naslov**.

```
45 $access_key = '3651d54581aee45468f595dd0aec5789';
46 $email_address = $user_email;
47
48 $ch = curl_init('http://apilayer.net/api/check?access_key='.$access_key.'&email='.$email_address.'');
49 curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
50
51 $json = curl_exec($ch);
52 curl_close($ch);
53
54 $validationResult = json_decode($json, true);
```

Slika 5.4: Izsek kode za pošiljanje http zahtevkov na spletni API.

Strežnik nato dobi odgovor na poslani http zahtevek. API vrne odgovor v obliki zapisa JSON. Odgovor poleg naziva elektronskega naslova vsebuje podatke o pravilnem zapisu elektronskega naslova in še nekatere druge podatke,

ki jih za potrebe naše aplikacije ne potrebujemo. Poleg pravilne oblike zapisa mora vsak elektronski naslov imeti vsaj še veljaven MX zapis.

## 5.7 Relacije med zaposlenimi in vnos spremnega besedila

V naslednjem koraku razvoja aplikacije smo se lotili izdelave sistema, kjer izbranemu uporabniku določimo skupino vprašalnikov in vse uporabnike, ki ga bodo ocenjevali po vprašalnikih, ki pripadajo izbrani skupini vprašalnikov. Uporabnike razvrstimo glede na relacije, v katerih so si z izbrano osebo. Izdelali smo krmilnik `RelacijeMedZaposlenimiController.php`, ki vsebuje logiko za črpanje in shranjevanje podatkov v podatkovno bazo. V pogled smo najprej dodali dva spustna seznama. V prvem spustnem seznamu se nahajajo vsi registrirani uporabniki brez administratorjev. V drugem spustnem seznamu pa se nahajajo vse trenutno vnesene skupine vprašalnikov. Poleg spustnih seznamov smo dodali še seznam vseh uporabnikov in sezname, ki predstavljajo posamezno relacijo in v katere dodajamo uporabnike iz seznama vseh uporabnikov.

Z uporabo ORM komponente smo iz podatkovne baze pridobili vse uporabnike. Nato smo spustne sezname v pogledih napolnili z uporabo funkcije `@foreach`, ki jo vsebuje `Blade`. Ko administrator v prvem spustnem seznamu izbere uporabnika in skupino vprašalnikov, se v drugem seznamu izvede AJAX klic na strežnik. Metoda `relacije_med_zaposlenimi` v krmilniku prejme identifikator izbranega uporabnika in skupine vprašalnikov ter naredi poizvedbo v podatkovno bazo. V tabeli `interviews` preverimo, če za uporabnika in skupino vprašalnikov že obstajajo zapisi in pridobljene podatke pošljemo nazaj v zapisu JSON. Nato z uporabo JavaScripta na strani odjemalca glede na podatke prejete v JSON odgovoru razvrščamo uporabnike v ustrezne sezname. Uporabnike iz seznama vseh zaposlenih lahko poljubno dodajamo v sezname sodelujočih z uporabo načina "drag and drop". "Drag and drop" način smo vključili z uporabo JavaScript knjižnice

jquery-ui. Ko zaključimo z urejanjem relacij, podatke shranimo s klikom na gumb za shranjevanje relacij. Preko AJAX klica na strežnik pošljemo podatke in jih shranimo v tabelo `interviews`.

Spremnno besedilo vnašamo preko urejevalnika besedila, ki smo ga vključili s pomočjo JavaScript knjižnice `Summernote`. V razdelek za vnos spremnega besedila smo dodali dva ločena urejevalnika besedil. V prvi urejevalnik vnašamo besedilo v prvi osebi ednine. To je besedilo, ki se pošlje uporabnikom, ki so povabljeni k ocenjevanju samega sebe. V drugi urejevalnik besedila vnašamo besedilo, ki se pošlje ostalim uporabnikom, ki so povabljeni k ocenjevanju. Spremna besedila smo shranili v podatkovno tabelo `email_text`. V spremno besedilo lahko poljubno vstavljamo posebne značke, ki se začnejo s posebnim simbolom `$`. Vse te značke pred pošiljanjem elektronskega sporočila nato z uporabo PHP funkcije `str_replace` nadomestimo z ustreznimi vrednostmi.

## 5.8 Urejanje vprašalnikov

Izdelave sistema za urejanje skupine vprašalnikov smo se lotili na podoben način kot sistema za določanje relacij med uporabniki. Dodali smo pot in ustvarili krmilnik. Zatam smo ustvarili še pogled in vanj dodali spustni seznam. Spustni seznam smo napolnili s skupinami vprašalnikov, ki smo jih pridobili iz podatkovne tabele `questionnairegroup`, kjer hranimo zapise o vnesenih skupinah vprašalnikov.

Poleg spustnega seznama smo dodali seznam vseh vprašalnikov in seznam, kamor s seznama vseh vprašalnikov z uporabo "drag and drop" metode vnašamo posamezne vprašalnike, ki bodo pripadali tej skupini vprašalnikov. Ob izbiri skupine vprašalnikov s spustnega seznama se naredi AJAX klic na strežnik. To smo storili tako, da smo z uporabo knjižnjice jQuery spustnemu seznamu dodali dogodek, ki se sproži z vsako izbiro vrednosti s spustnega seznama. Nato smo napisali še JavaScript funkcijo, ki jo sproži dogodek in poskrbi za pošiljanje http zahtevkov na strežnik.

Na strežnik pošljemo podatke o izbrani skupini vprašalnikov in pripadajočih

vprašalnikih. Na strežniku nato v podatkovno tabelo shranimo poslane podatke. V tabeli `questionnairegroupquestionnaire` hranimo vse zapise o tem, katere vprašalnike vsebuje posamezna skupina vprašalnikov.

## 5.9 Ocenjevanje

Vsakemu uporabniku, ki je bil povabljen k ocenjevanju določene osebe, se za vsako ocenjevanje na elektronski naslov pošljejo povezave preko katerih uporabniki dostopajo do vprašalnikov.

Povezave za posamezne vprašalnike sestavimo iz zapisov v tabeli `interview`. S klikom na povezavo uporabnik prispe do vprašalnika. Vse poti za ocenjevanja smo zavarovali z vmesnima mehanizmom `Auth` in `Interview`. Prvi vmesni mehanizem onemogoči dostop vsem neprijavljenim uporabnikom. Drugi vmesni mehanizem pa za uporabnika, ki prispe na povezavo v podatkovni tabeli `interview`, preveri, če je bil uporabnik povabljen na ocenjevanje tega vprašalnika. Če v tabeli ni ustreznih zapisov, se uporabnika preusmeri na nadzorno ploščo uporabnika, kjer ima prikazan seznam vseh povabil na ocenjevanje. Vsakemu vprašalniku v seznamu smo dodali tudi gumb, ki uporabnika preusmeri na stran za ocenjevanje.

V krmilnik, ki vsebuje logiko za izvedbo ocenjevanja, smo dodali metodo `interview`, ki prejme parametre s spletnega naslova. Vse parametre, ki jih pošiljamo po spletnem naslovu, smo iz varnostnih razlogov zamaskirali z uporabo zgoščevalnih funkcij Laravel komponente `Hashids`, ki smo jo namestili z uporabo orodja Composer. Metoda `interview` nato iz prejetih parametrov iz podatkovne baze pridobi vprašanja in jih razvrsti po vprašalnikih, ki jim pripadajo. Povezovalna logika nato vprašanja porazdeli po posameznih kategorijah in kompetencah. V pogledu smo z uporabo skriptnega jezika JavaScript ustvarili uporabniški vmesnik za odgovarjanje na vprašanja.

## 5.10 Izdelava poročil

Zadnji korak pri razvoju aplikacije je bila izdelava sistema za izdelavo poročil za uporabnike, ki so bili ocenjevani. V ogrodju smo najprej ustvarili vse potrebne PHP razrede, ki vsebujejo logiko za preračun in prikaz podatkov v poročilu. Vsako poročilo se prične z naslovno stranjo, ki vsebuje logotip in naziv podjetja ter ime in priimek ocenjevane osebe. Naslovni strani nato sledi stran, ki vsebuje oštevilčeno kazalo vsebine. Vsebinsko je posamezno poročilo razdeljeno na pet večjih poglavij.

V prvem poglavju je najprej na kratko predstavljeno ocenjevanje po metodi 360 stopinj. V drugem poglavju sledi predstavitev vsebine poročila. Ta vsebuje statistične podatke o ocenjevanju za posameznega uporabnika in kratko razlago o vsebini poročila. V naslednjem poglavju je z uporabo grafov predstavljen pregled rezultatov po posameznih kompetencah.

Grafični prikaz rezultatov smo naredili z uporabo JavaScript knjižnice Highcharts. Poleg grafičnega prikaza se v tretjem poglavju nahaja še celosten pregled izbranih kompetenc, ki je prikazan z uporabo HTML tabel. V četrtem poglavju sledi podrobnejši pregled posameznih kompetenc.

Z uporabo HTML tabel smo prikazali podrobnejši statistični prikaz za vsako kompetenco in kategorijo ter nato še podrobnejši statistični prikaz ocenjevanja za posamezno vprašanje določene kategorije. V zadnjem poglavju se nahajajo dodatna vprašanja, na katere udeleženci odgovarjajo, ko prejmejo poročila.

Poročila, ki morajo biti prirejena za tisk v A4 formatu, smo izdelali z uporabo posebnih medijskih poizvedb jezika CSS in uporabo označevalnega jezika HTML. Po izdelavi posebnih razredov in metod, ki iz podatkovne baze črpajo podatke in ustrezno pripravijo podatke za prikaz, smo ustvarili še pogled, ki vsebuje vso HTML kodo. HTML oznakam smo dodali razrede in identifikatorje ter jih tako povezali z medijskih poizvedbami. Nato smo z uporabo funkcij jezika Blade znotraj HTML oznak vstavljali pripravljeno vsebino. Poročilo se uporabniku prikaže v obliki običajne spletne strani. Z

uporabo funkcije `Natisni`, ki jo vsebujejo spletni brskalniki, se naredi izpis poročila v obliki, ki je primerna za tisk in smo jo določili s posebnimi medijskimi poizvedbami jezika CSS.

```

1  /** PRINT **/
2  @page
3  {
4      margin: 15mm 15mm 15mm 15mm; /*size: A4; */
5  }
6
7  @page:first {
8      @bottom { content: normal; font-style: italic; border-top: none; }
9      @bottom-right { content: ""; border-top: none; }
10     @bottom-left { content: ""; border-top: none; }
11     @footnotes { border-top: none; }
12 }
13
14 @page:right {
15     @bottom-right { border-top: .25pt solid #666; }
16     @bottom-left { margin: 10pt 0 30pt 0; border-top: .25pt solid #666; content: ""; }
17     @bottom-right { margin-top: 7pt; border-top: .25pt solid #666; font-size: 9pt; }
18     @bottom-left { margin-top: 7pt; border-top: .25pt solid #666; font-size: 9pt; }
19     @bottom { margin-top: 7pt; border-top: .25pt solid #666; font-size: 9pt; }
20 }
21
22 @page:left {
23     @bottom-left { content: counter(page) "/" counter(pages); }
24     @bottom-right { margin: 10pt 0 30pt 0; border-top: .25pt solid #666; content: ""; }
25     @bottom-right { margin-top: 7pt; border-top: .25pt solid #666; font-size: 9pt; }
26     @bottom-left { margin-top: 7pt; border-top: .25pt solid #666; font-size: 9pt; }
27     @bottom { margin-top: 7pt; border-top: .25pt solid #666; font-size: 9pt; }
28 }
29
30 div.page{
31     page-break-after: always;
32     page-break-inside: avoid;
33 }

```

Slika 5.5: Medijske poizvedbe za tisk v stilskih slogovnih predlogah.



## Poglavje 6

### Sklepne ugotovitve

V sklopu diplomske naloge smo razvili spletno aplikacijo za ocenjevanje po metodi 360°. Razvoj spletne aplikacije smo pričeli z določanjem primerov uporabe aplikacije. Ko smo določili vse potrebne funkcionalnosti aplikacije, smo pričeli s konceptualnim načrtovanjem podatkovne baze. Sledila je izbira orodij in tehnologij, s katerimi smo razvijali spletno aplikacijo. Odločili smo se za uporabo ogrodja Laravel, ki uporablja MVC arhitekturo in vsebuje veliko komponent, ki so nam olajšale razvoj posameznih funkcionalnosti aplikacije. Vzpostavili smo sistem za registracijo in prijavo uporabnikov. Poleg sistema za avtentikacijo uporabnikov smo razvili sistem za avtorizacijo uporabnikov. Aplikacija vsebuje dvouporabniško strukturo. Uporabniki imajo lahko vlogo skrbnika ali navadnega uporabnika. Glede na posamezno vlogo, ki je ob registraciji dodeljena uporabnikom, je bilo potrebno razviti sistem, ki uporabnikom omogoči dostop le do tistih razdelkov aplikacije, ki so dodeljeni za posamezno vlogo. Sistem za avtorizacijo uporabnikov smo razvili z uporabo vmesnih mehanizmov, ki jih vsebuje ogrodje. Aplikacija omogoča preprost vnos vseh potrebnih podatkov za izvedbo ocenjevanja. Uporabniški vmesnik je prilagojen za prikaz tako na namiznih kot tudi mobilnih napravah. Razvili smo tudi sistem za kreiranje in tisk poročil, kar smo storili z uporabo posebnih medijskih poizvedb jezika CSS. Medijske poizvedbe za tisk smo morali prilagoditi za uporabo na različnih spletnih brskalnikih. Izpolnili smo vse

cilje, ki smo si jih zadali pred pričetkom razvoja aplikacije. Največ časa smo porabili za razvoj sistema za kreiranje poročil. Vsa ustvarjena poročila za posameznega uporabnika morajo biti pripravljena za tisk v A4 formatu. Vsebovati morajo naslovno stran, oštevilčeno kazalo vsebine, oštevilčene strani in nogo ter glavo na posamezni strani. Stroškovno in časovno učinkovitost smo poskušali doseči z uporabo odprtokodnih ogrodij. Z uporabo ogrodja Laravel smo privarčevali veliko časa pri razvoju zalednega sistema. Prav tako je aplikacija zasnovana modularno in omogoča hiter razvoj novih funkcionalnosti in urejanje že obstoječih. Po razvoju in testiranju aplikacije je bila aplikacija uspešno uporabljena za ocenjevanje zaposlenih v podjetju Špica international, d. o. o.





# Literatura

- [1] Composer. Dosegljivo: [http://sl.phptherightway.com/#composer\\_and\\_packagist](http://sl.phptherightway.com/#composer_and_packagist). [Dostopano: 15. 1. 2017].
- [2] Kaj je CSS? Dosegljivo: <http://nsa-splet.si/css/zgledi/css-02-kaj-je-css.php>. [Dostopano: 5. 3. 2017].
- [3] Kaj je PHP? Dosegljivo: <http://nsa-splet.si/php/php.php>. [Dostopano: 28. 2. 2017].
- [4] Kaj je HTML? Dosegljivo: <http://www2.arnes.si/~acimpr/ROM/HTML/osnovehtml.html>. [Dostopano: 15. 1. 2017].
- [5] Janez Hudovernik. Odkrijte analizo kompetenc po metodi 360°. 2015.
- [6] JavaScript in HTML? Dosegljivo: <http://zaversnik.fmf.uni-lj.si/gradiva/javascript/uvod/script.php>. [Dostopano: 10. 1. 2017].
- [7] Laravel. Dosegljivo: <https://laravel.com/docs/5.2>. [Dostopano: 25. 2. 2017].
- [8] Email Validation and VerificationJSON API for Developers. Dosegljivo: <https://mailboxlayer.com/>. [Dostopano: 5. 3. 2017].
- [9] Metoda 360 stopinj. Dosegljivo: [https://sl.wikipedia.org/wiki/Metoda\\_360\\_stopinj](https://sl.wikipedia.org/wiki/Metoda_360_stopinj). [Dostopano: 3. 3. 2017].
- [10] SQL. Dosegljivo: [https://developer.mozilla.org/en-US/docs/AJAX/Getting\\_Started](https://developer.mozilla.org/en-US/docs/AJAX/Getting_Started). [Dostopano: 28. 2. 2017].

- 
- [11] MVC. Dosegljivo: <https://blog.codinghorror.com/understanding-model-view-controller/>. [Dostopano: 25. 2. 2017].
- [12] Dialogos, 360 ° povratna informacija: sebe razkrivamo skozi druge. Dosegljivo: <http://www.dialogos.si/slo/objave/clanki/povratna-informacija/>. [Dostopano: 4. 3. 2017].
- [13] PowerDesigner. Dosegljivo: <https://en.wikipedia.org/wiki/PowerDesigner>. [Dostopano: 5. 3. 2017].
- [14] SQL. Dosegljivo: <https://sl.wikipedia.org/wiki/SQL>. [Dostopano: 28. 2. 2017].
- [15] XAMPP. Dosegljivo: <https://en.wikipedia.org/wiki/XAMPP>. [Dostopano: 20. 2. 2017].